



PERGAMON

Information Processing and Management 37 (2001) 53–74

www.elsevier.com/locate/infoproman

**INFORMATION  
PROCESSING  
&  
MANAGEMENT**

## MEMOIR — an open framework for enhanced navigation of distributed information

D. De Roure<sup>a,\*</sup>, W. Hall<sup>a</sup>, S. Reich<sup>a</sup>, G. Hill<sup>b</sup>, A. Pikrakis<sup>c</sup>, M. Stairmand<sup>d</sup>

<sup>a</sup>*Multimedia Research Group, University of Southampton, Southampton SO17 1BJ, UK*

<sup>b</sup>*Multicosm Ltd, Southampton, UK*

<sup>c</sup>*Department of Informatics, University of Athens, Greece*

<sup>d</sup>*IT Innovation, Southampton, UK*

Received 30 March 1999; accepted 3 March 2000

---

### Abstract

In large companies, whose business is critically dependent on the effectiveness of their R&D function, the provision of effective means to access and share all forms of technical information is an acute problem. It is often easier to repeat an activity than it is to determine whether work has been carried out before.

In this paper we present experiences in implementing and evaluating the MEMOIR system. MEMOIR is an open framework, i.e., it is extensible and adaptable to an organization's infrastructure and applications, and it provides its user interface via standard Web browsers. It uses trails, open hypermedia link services and a set of software agents to assist users in accessing and navigating vast amounts of information in Intranet environments. Additionally, MEMOIR exploits trail data to support users in finding colleagues with similar interests. The MEMOIR system has been installed and evaluated by two end-user organizations. This paper describes the results obtained in this evaluation. © 2000 Elsevier Science Ltd. All rights reserved.

*Keywords:* User trails; Open hypermedia framework; Software agents; Evaluation; Corporate memory

---

---

\* Corresponding author. Fax: +44-23-8059-2865.

*E-mail address:* dder@ecs.soton.ac.uk (D. De Roure).

## 1. Introduction

Finding people with similar interests can be as difficult as efficiently accessing documents with related information. This is especially true for research oriented, globally distributed organizations such as pharmaceutical companies. It is often easier to repeat work than it is to determine whether work has been carried out before. The MEMOIR project (Managing Enterprise-scale Multimedia using an Open Framework for Information Re-use (Hill et al., 1997)) addresses these problems by supporting the user with extra assistance provided by a set of software agents that mine *trailbases* and *linkbases*.

A user's trail is the set of actions on documents that they have visited (so-called *trailmarks*) in pursuing a certain task. Users with similar interests follow similar paths and therefore they leave comparable trails. By matching trails, we match users, i.e., the similarity measures are based on traversal only (see also Tudhope and Taylor (1997) for other measures). MEMOIR lets the user ask questions such as “who else reads similar documents?” and “what else should I read?”.

In addition to trails, MEMOIR makes use of advanced hypertext technology such as open hypermedia link services in order to allow the users to manage associations between different types of documents in an independent and flexible way (De Roure, Hall, Davis & Dale, 1996; Pikrakis et al., 1998).

This paper presents an overview of MEMOIR and discusses the experiences of implementing and evaluating the system. The paper is structured as follows. The following Section 2 outlines the open framework that the system is built upon. In Section 3 we investigate related work. Section 4 then describes our experiences both from a technical point of view in constructing the system and from an end-user's point of view in actually using the system. We finish with a summary and conclusions in Section 5.

## 2. The MEMOIR system

The development of MEMOIR was driven by both the end-users' requirements (Hill et al., 1997; Loades, James, Hutchings & Stairmand, 1997; Malcolm, Poltrock & Schuler, 1991) and general criteria for open hypermedia systems which have been discussed extensively at Open Hypermedia Systems Workshops (Wiil, 1997, 1999).

While open hypermedia systems treat hypermedia links as first class objects, MEMOIR additionally promotes another kind of object: the trail. The notion of trails dates back to the very first ideas of hypertext, as expressed by for example *Bush's* “Memex” (Bush, 1945) or *Englebart's* “Augment” (Engelbart, 1963). Bush envisaged trails (or paths (Bieber, Vitali, Ashman, Balasubramanian & Oinas-Kukkonen, 1997; Furuta, Shipman, Marshall, Brenner & Hsieh, 1997; Shipman, Furuta, Brenner, Chung & Hsieh, 1998; Zellweger, 1989), or footsteps (Nicol, Smeaton & Slater, 1995) as they are called elsewhere) in order to provide a mechanism for managing a user's personal information but also to allow for these trails to be available to other users. Bush (1945) envisaged a “new profession of trail blazers, those who find delight in the task of establishing useful trails through the enormous mass of the common record”. MEMOIR aims at providing a framework for the upcoming applications that will be used by

trail blazers. It treats trails as ‘first class’ objects: trails are objects in their own right and can be manipulated by agents in the system.

Reading (i.e., opening and viewing) a resource, which is available via Web browsers, might be the most common type of activity. However, there are many more interesting activities users perform: printing a document could indicate a special interest or a high priority; mailing a document to someone else might have similar indications; creating a link between two documents might indicate an important relationship (for activity-centred information access (see also Chalmers, Rodden & Brodbeck, 1998; Plaisant, Shneiderman & Mushlin, 1998)). Rosenberg (1996) provides a framework for hypertext activities. However, because of the restrictions of current Web browsers (Bieber & Wan, 1994; Grønbaek, Bouvin & Sloth, 1997; Jones & Cockburn, 1996; Tauscher and Greenberg, 1997) MEMOIR currently considers only a subset of this framework, in particular activities such as ‘open document’; conceptually the system could also trace other activities such as printing a document or even creating a trail. In this way the management of trails by users themselves might become part of their trail.

Defining the term agent has been — and still is — a controversial and unfinished task (Huhns & Singh, 1997; Nwana & Ndumu, 1996; Petrie, 1997; Shneiderman and Maes, 1997). For the purposes of this paper, we define an agent to be a software entity which responds to a request made by a user or by another agent. The problem domain that we address is Distributed Information Management, hence we call these agents *DIM* agents. In fact, we tackle DIM applications from a weak agent’s perspective (Wooldridge & Jennings, 1995): our agents are autonomous and reactive, sometimes they are proactive (see Section 4.1.1). In addition to resource discovery, we emphasize the role of DIM agents in resource maintenance and integration (De Roure et al., 1996).

Rather than aiming at supporting collaboration in Intranet environments by sharing documents (see e.g., the BSCW system (Bentley et al., 1997) or also the WEBDAV Initiative (Whitehead, 1997)) we provide agents that analyse the trails users store as well as the documents that users read (Dodge, Marx & Pfeiffenberger, 1995). Conceptually, the activities of which trailmarks are composed can also be processed (Twidale, Nichols & Paice, 1997). Given information about trails, agents are able to provide users with information about other users in the system with similar interests. The similarity is based on the trailmarks contained in a trail, the keywords contained in these trailmarks and also the user’s personal profile which is stored as a set of weighted keywords. Further details on the MEMOIR agent system can be found in Pikrakis et al. (1998).

### 2.1. *The architecture*

The MEMOIR architecture is an open architecture that can be dynamically extended and adapted to an organization’s infrastructure and applications. The architecture is an evolution of the Distributed Link Service (Carr, De Roure, Hall & Hill, 1995). The DLS is a ‘third party’ link service (the link service provider may be distinct from the content provider) employing one or more link servers which can be interrogated either before or after a document is viewed. To support a button-oriented style of interaction (as is known for example from the Web), the link service is interrogated in advance and links are inserted into the document before delivery to the user. The user requires no more than a standard Web browser.

The link intersection can occur in batch, so that the links are ‘compiled in’ before the documents are served, or on-the-fly as the document is delivered. The proxy DLS (or DLS Agent) implements the on-the-fly scenario.

MEMOIR extends the DLS to support collaboration by storing and processing trails. It introduces new system components: a distributed object-oriented database, a framework for agents, and a communications message router. Fig. 1 gives an overview of the different components and their interaction. Components displayed in dashed lines are third party components; interactions using other than the HyperText Transfer Protocol (HTTP) as a protocol are drawn using dashed arrows.

The key component is the *message router* which acts as a hub with which other system services register. Its design is based on Microcosm’s filter model (Hill, Wilkins & Hall, 1993) and the subsequent ‘TNG’ model for distributed hypermedia link services (Goose, Dale, Hill, De Roure & Hall, 1996). Any component makes its services available by registering details of its services with the message router, which in turn can route particular requests from other components to the newly registered one. Messages for querying all currently available services are part of the set of message router services. The model of servicing a component allows the system to be dynamically tailored to specific needs. New services can be easily added, and multiple providers for the same service are possible. Multiple linkbases, for example, could be managed by one component providing services for link manipulation; they could equally be managed by different components providing the same services. Multiple message routers are possible.

Distribution is addressed at various levels. Firstly, all components can be run on different hosts. Secondly, the use of the HTTP allows components on heterogeneous platforms to communicate. Thirdly, the underlying object-oriented database system ITASCA (see below) can be distributed and by that provide the means for sharing data.

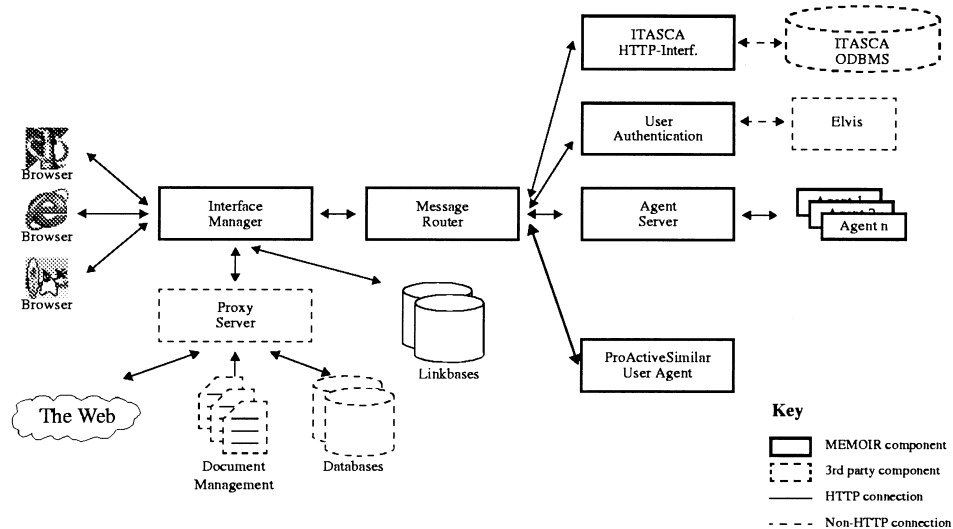


Fig. 1. MEMOIR components.

Standard Java enabled Web browsers provide the *user interface* to the MEMOIR system in order to minimize installation and maintenance overhead. The base panel shown in Fig. 2 is a Java applet that provides the user interface (see also Section 2.2. below). The only configuration necessary is to set the interface manager as a proxy server and to connect once to it in order to log in. As browsing activity progresses the trail display in the user interface is updated, and users may select documents from the trail as input to MEMOIR functionality. For instance they might choose to select some trail marks, press the button named ‘suggest reading for documents’ and the results would be returned in the lower list box of the base panel applet. The applet serves as an interface to the following functionality:

- link management, including creation of links and link navigation;
- trail management, including ‘finding similar person by trail’, querying for suggestions of other documents to be read, and trail creation;
- keyword services, such as extracting keywords from a document as well as finding users or documents based on keywords.

The *interface manager* behaves as an HTTP proxy server and hence is able to listen to a user’s requests and commands. It can itself be configured to use another proxy (e.g., cache or firewall). It is via the interface manager that users view the Web, document management systems and also access linkbases. Thus, existing data can be re-used and is not stored within

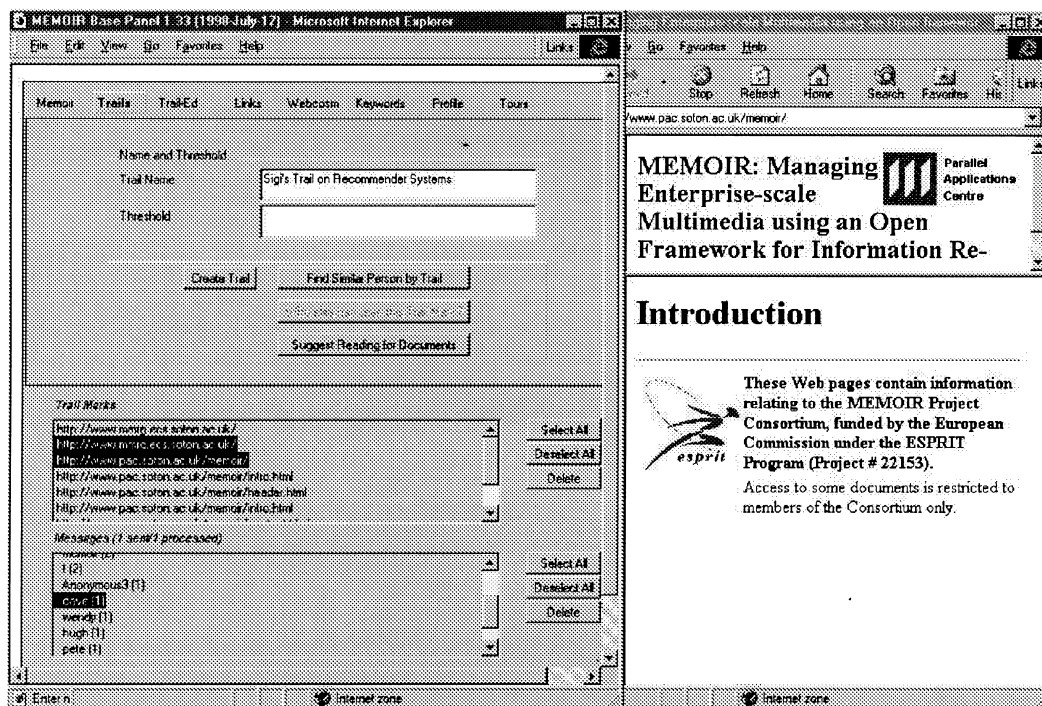


Fig. 2. Basic control panel and browsing window.

the system (though meta data such as a document's keywords are extracted and stored in the data repository, ITASCA).

An *authentication component* is responsible for managing user authentication. As far as possible this component relies on existing user management systems.

A dedicated *agent server* manages a set of software agents and provides different agent services that are essentially responsible for data mining and resource discovery tasks such as searching the Web (Pikrakis et al., 1998). It serves as an agent service broker and as a gateway for messages directed to simple or complex agents from various components. Individual agents themselves may also autonomously contact other agents or access resources available on the Web.

The ITASCA distributed object oriented database system (IBEX, 1995) is used as an underlying data repository. Additionally it provides the framework for a number of agents that are implemented directly in LISP, the data manipulation language of ITASCA. As HTTP is used as communication protocol, an interface component to ITASCA is provided allowing the communication with the database via HTTP (see Fig. 1).

During the design phase of MEMOIR a variety of alternatives for message passing (including CORBA/IIOP and proprietary systems such as IBM's MQ Series) were considered. One of our design goals was to provide a system which would integrate with the existing computing infrastructure with minimal support impact. Hence, we chose the HTTP as a communication layer protocol because it is a widely supported standard and lightweight protocol.

We distinguish between two basic forms of communication, namely notifications and requests. These are expressed using POST or GET methods respectively, a convenient mapping which sits comfortably in existing Web infrastructure.

- Where a process simply wishes to provide notification that an event has occurred, such as the opening of a document, the POST request can be used to deliver a message of a particular type to the message router. The message router can in turn report the event to any processes registered to receive messages of this type using further notifications.
- For queries or requests where a reply is required, such as following a link, the GET request is used. As with POST requests, the message is passed on to the appropriate components, and all results can be packaged together to be returned to the calling component.

The MEMOIR message format is based on the open, extensible structure used by Microcosm (Davis, Hall, Heath, Hill & Wilkins, 1992). This format uses simple tag/value pairs and although there is a required set of tags for normal operation of the system, additional tags can easily be added to extend the information carried to suit new functions. Due to the availability of this protocol, the fact that MEMOIR is used in a Web environment, and the fact that the protocol supports the necessary primitives and is extensible, we refrained from using agent communication languages such as KQML (Labrou & Finin, 1997) or ACL (FIPA, 1997) though we have adopted these in other projects.

Messages can be either synchronous or asynchronous. It has proven very useful to support an asynchronous model in addition to the synchronous client-server model imposed by the Web architecture. Notifications enable the system to exhibit proactive behaviour and avoid the need to maintain processes that poll for state changes of other components or remote objects.

Some operations do not need to occur in realtime. This permits, for example, the user interface to group a user's requests and answers together, identifying those that refer to the same ID. If a result is returned while the user is not logged on to the system, it is stored in a message box within the ITASCA database and retrieved upon restart of the component.

## 2.2. Working with MEMOIR

It is up to the user to decide what, i.e., which trail marks, should make a trail; it is also important to mention that trails are only *displayed* to the user automatically but are not *stored* automatically (in order to preserve privacy, see Section 4.2.3.2). Thus, the users themselves benefit by keeping track of places they have visited. If they want to share this information, however, they may explicitly select and store the trails and in this way make them available to the system, i.e., the agents and the other users.

Profiles are used in order to allow people to work with MEMOIR in different roles (Baudisch, 1997; Belkin & Croft, 1992; Gottlob, Schrefl & Röck, 1996). Users might change profiles dynamically; it is also possible to have more than one profile selected. Each user action such as querying the trailbase or creating a link takes the currently selected user profiles into account. Thus, users querying with a 'sales profile' selected will get different results from doing the same request with a 'research profile' selected.

Many recommender systems suffer an incentive problem (Resnick & Varian, 1997) (also called the *cold start problem* (Maltz & Ehrlich, 1995)); the system is only as good as the data in it. MEMOIR addresses this problem by introducing a priming phase where data from existing proxy servers is used, from which trails are automatically generated. For reasons of privacy we only provide an 'anonymous' user name. Hence, during this introductory phase the system is able to provide answers to the effect that 'someone' has already visited a certain site.

## 3. Related work

The MEMOIR system shares similarities with several active areas of research and development, which may be loosely grouped together under the headings of:

- information filtering systems;
- systems based on the users' browsing behaviours; and
- recommender systems.

Note, that some of the systems described in the following might fall into more than one category, so these categories are overlapping. However, we describe the systems within the category that we believe they fit best.

### 3.1. Information filtering systems

Information filtering systems apply different algorithms such as pattern matching with contextual analysis in order to provide the users with information that they have specified in a

textual way. Mostly news services are used as input. Users define their preferences by giving keywords or voting on pages.

*Agentware i3* (Autonomy, 1997) for instance, is a typical representative of this category of systems. Based on information about the users' interests, which they provide by entering keywords, it applies pattern matching algorithms with contextual analysis to some 50 new services in order to provide users with relevant documents.

*NetAttaché Pro* and *Atlas Server* (Tympani, 1997) are Web agents that retrieve and filter Web data (also in off-line mode) and allow for additional searches based on user preferences. A search engine front-end supports a number of Internet search engines and news services, it allows for the import of bookmark and other link files and additionally users can specify their own Boolean queries. The *Atlas Server* supports multiple users by employing search and filter agents to a group of users' browsing data.

*Beehive* (Huberman & Kaminsky, 1996) is a prototype system supporting cooperative filtering and sharing by automatically recording interactions within a community. A user sets up a number of communities and associated e-mail addresses and the system then automatically monitors interactions between members of communities. These communities can be shared and in this way users can be informed e.g., about changes of a URL.

*Yenta* (Foner, 1996) is a prototype multi agent system that aims at matching people with similar interests and in this respect shares similarities with MEMOIR. A user's interests are determined by analyzing news messages, e-mails and personal documents; the author claims though that the system is flexible and thus able to work on any type of electronic data given that there is a way to measure similarity for it. Systems within this category mainly suffer the problem that users manually, often with a lot of effort, have to express and maintain their profiles. Trail based systems on the other hand, can provide a higher degree of automation in that users express their interests simply by the documents they are viewing and publishing as trails.

### 3.2. Systems based on the users' browsing behaviours

This category of systems provides tour guide agents that accompany users from page to page providing assistance in browsing/navigating. The agent's knowledge is based on learning from previous tours and hypertext structure.

IBM's *Web Browser Intelligence* (WBI) (Barrett, Maglio & Kellem, 1997) for instance, is a proxy based system that allows agents to process the information that has been seen by a user. WBI is programmable and can be described as an extensible proxy server shell for personalizing Web data.

*Vrisko* (Zuno, 1997) is a personal knowledge manager client implemented in Java. It provides facilities such as integrated searching over many search engines and relevance ranking of results. Vrisko allows users to specify a context for the search; also the users' bookmarks can be re-used. Vrisko allows for user profiling. The profiles are updated according to user feedback and also automatically by search results. Automatic keyword searches can only be done on textual documents.

*WebWatcher* (Joachims, Freitag & Mitchell, 1997) is a tour guide agent that accompanies users from page to page providing assistance in browsing/navigating. The agent's knowledge is



based on learning from previous tours and hypertext structure. Additionally, the user can communicate with the system and give feedback. During the development of WebWatcher the main focus has been on its learning capabilities and on the user interface.

*Letizia* (Lieberman, 1995) is a user interface agent that assists users in browsing the Web. The system provides users with suggestions of places to visit based on the user's previous browsing behaviour. As *Letizia* is implemented for a specific platform (in this case the Apple Macintosh) it is able to retrieve information about other than 'view document' events. For instance, storing a bookmark, saving a document, etc., can be traced.

*BRIO* (Röscheisen, Mogensen & Winograd, 1995) is a general architecture and its implementation that deals with annotations on the Web. The system also allows for the manual editing of trail marks. These trail marks can be used for navigational purposes.

The *Do-I-Care* collaborative Web agent (Starr, Ackerman & Pazzani, 1996) aims at supporting collaboration by providing users with information about changes to relevant documents. The system claims to avoid users having to do extra work. Relevant documents are detected by keeping a list of URLs that a user has visited together with the number of visits. Changes to documents are detected by computing mutual information and checking changes in size and number of links. Users are notified about the changes by e-mail. Collaboration is supported by sharing information about changes.

Other systems within this category include the *Knowledge Pump* (Glance, Arregui & Dardenne, 1997) and some systems of the group of recommender systems (Resnick & Varian, 1997).

We see the main deficiencies of these systems with respect to MEMOIR in the following facts. Firstly, none of the systems mentioned implements user trails as 'first class' objects, i.e., objects that can be created, manipulated, copied, etc. Some re-use paths followed by other users (Barrett, Maglio & Kellem, 1997; Joachims et al., 1997; Kahle & Gilliat, 1997; Resnick & Varian, 1997) or edited explicitly (Röscheisen et al., 1995). We believe the management of trails as first-class objects to be a key feature of trail based systems. Similar to the way users handle URLs today, we envisage trails being passed on, sent via e-mail and the like. Furthermore, few systems support the idea of an open framework that can be easily extended or changed (notable exceptions are *Autonomy* 1997; Barret et al., 1997). Generally, the majority of systems focus on agents serving as personal knowledge managers rather than on frameworks that support collaboration amongst users.

### 3.3. Recommender systems

Recommender systems (Resnick & Varian, 1997) (sometimes also referred to as collaborative filtering or social filtering systems (Goldberg, Nichols, Oki & Terry, 1992)) exemplify a category of systems that are based on the idea that users often face the problem of having to make choices without sufficient experience and therefore rely on other people's recommendations. Recommender systems assist this process. Rather than using user trails as a main source for providing assistance, recommender systems have intelligent agents for weighing user votes, for instance, Web pages as done by *Alexa* (Kahle & Gilliat, 1997), or usenet articles in general (Falk & Jonsson, 1996; Hattori, Ohguro, Yokoo, Matusbara & Yoshida, 1999; Konstan et al., 1997; Terveen, Hill, Amento, McDonald & Creter, 1997), for music albums and

artists as done by *Ringo* (Shardanand & Maes, 1995) or based on e-mail (Huberman & Kaminsky, 1996).

*Alexa* (Kahle & Gilliat, 1997) is a navigational service that gives the user recommendations based on other users' browsing experience as well as explicit "Where to go next" recommendation that other users can add. *Fab* uses both user votes and content based filtering techniques (Balabanović & Shoham, 1997). *WebTagger* (Keller, Wolfe, Chen, Rabinowitz & Mathe, 1997) and *SOaP* (Guo, Kreifelts & Voss, 1997) base their recommendations on bookmarks.

*Ringo* (Shardanand & Maes, 1995) is a system for personalized recommendations for music albums and artists. It is based on an approach called *social information filtering*. This approach essentially automates the word-of-mouth recommendations by weighing user votes with different algorithms. Users contribute by assigning votes on a scale from one to seven. The system maintains user profiles that record a user's positive and negative interests. These profiles are compared and recommended to the users. *Ringo* is the foundation for the *Firefly System* (see <http://www.firefly.com/>).

The *Vortex* system (Dieberger, 1997) addresses 'social navigation' by supporting the management of URLs in an advanced way, i.e., by allowing users to communicate directly and to interact directly with other users and in that way share information and provide assistance in selecting appropriate targets for navigation.

MEMOIR mainly differs from these systems by the following facts. Firstly, it aims at finding *users* rather than *documents* (in doing so it shares similarities with e.g., *Yenta* (Foner, 1996)). Secondly, MEMOIR treats user trails and links as 'first-class' objects, i.e., they can be created, updated, forwarded, etc. similar to the way users manage URLs today. Furthermore, MEMOIR is an open and flexible system based on the Web's infrastructure which few of the above mentioned systems are (notable exceptions are e.g., *Autonomy*, 1997; Terveen et al., 1997).

#### 4. Experiences

In this section we will report on the experiences in MEMOIR. We have split the section into two main parts. The first part focuses on our experience of using the MEMOIR framework in a number of research projects; the second part will describe the actual end-user evaluation.

##### 4.1. MEMOIR as a research vehicle

The MEMOIR framework and components have been used as a research vehicle in a number of research projects. The purpose of this was twofold. On the one hand we were aiming at enhancing the functionality, drawing on the expertise available at the University project partners. On the other hand we wanted to demonstrate the openness and re-usability of the framework.

###### 4.1.1. Dynamic extensibility

A proactive 'similar user' agent has been developed as an additional agent service in order to

show the dynamic extensibility of MEMOIR. The aim of the proactive agent was to inform a user actively about similar users, rather than having the user asking the system for this information. Fig. 3 shows the user interface of this agent.

While pages are loaded and displayed to the user, in the background this agent asks other agents in the system for basic information about users and their interests in the page currently being viewed. It then presents its result in the user interface and, if it has found other users, announces that ‘similar users’ have been found. Usernames, e-mail addresses and telephone numbers of the users are shown.

Several lessons have been learned while doing this experiment. Firstly, the experience clearly showed the dynamic extensibility of the MEMOIR system. The additional agent simply registers its service with the message router at runtime and in this way extends the available functionality. Furthermore, the benefits of using proxy processes for extending the functionality available at the browser’s side without additional reconfiguration were proven.

#### 4.1.2. Guided tours using trails

Having user trail information available, one obvious application is guided tours, and we have implemented such a mechanism in another project (Andric, Griffiths, Reich, Davis & Hall, 1998). For reasons of proof-of-concept, the tours are stored in a separate commercial

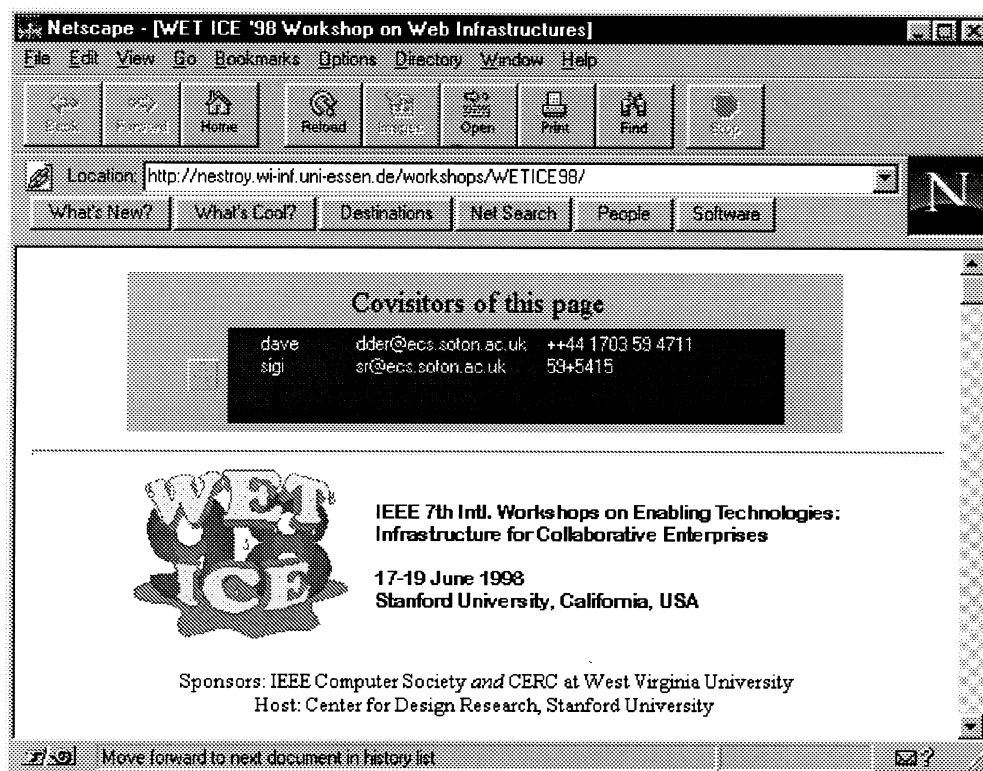


Fig. 3. Proactive agent.

database (MS-Access) and not in our main information repository (ITASCA). The user interface is implemented as an additional tabbed folder (see Fig. 2); implementing it as a separate window, such as in Jühne, Jensen and Grønbaek (1998), would have been possible as well but we did want to keep one control panel for all functionality.

This second example again demonstrates the extensibility and flexibility of the MEMOIR framework. It also shows that trail information can be re-used in various different ways.

#### 4.2. End-user evaluation

The MEMOIR prototype has been evaluated by the two corporate end-users participating in the project, namely Glaxco Wellcome (UK) and Unichema (NL). The subgroups doing the evaluation within these organizations include researchers that typically need to browse and search large numbers of documents prior to selecting those they wish to read.

Within one organization we had a user group of approximately 10 expert users trialling the system over a period of about three months in total. This group mainly served the purpose of giving first feedback, finding bugs, and testing the platform independence of the user interface. Our second group of end-users comprised 22 people in total. This group performed the evaluation within the normal day-to-day operations, i.e., there was no specific time dedicated to it.

The evaluation aimed at supporting a broad analysis of the MEMOIR system, including the dimensions shown below. Thus, the evaluation shares similar objectives with e.g., SUE (Systematic Usability Evaluation (Garzotto & Matera, 1997)). We did use a different methodology however, named SUMI (Software Usability Measurement Inventory (Kirakowski & Corbett, 1993), details see below). The following dimensions were investigated during the evaluation.

- Usability. The evaluation should determine what impact the system has on a user's day-to-day work, what onus it places on users, whether it is intuitive to use and whether the interface to services is appropriate.
- Functionality. The evaluation should determine that the system works, and that it works as users would expect. It is crucial to establish whether the functionality is useful to evaluators. Performance and robustness are other criteria of this dimension.
- Knowledge Management Objectives. The evaluation should determine to what extent the MEMOIR system realizes the Knowledge Management-based business benefits envisaged at the outset of the project.

The data for measuring these dimensions was mainly derived from questionnaires, log files of system usage and interviews.

The evaluators were given introductory training sessions in which the concepts as well as the user interface of MEMOIR were explained. The users then were free to use the system. The results reported below were obtained from the feedback received in an overall usage of the system in a period of about three months. As mentioned, about 30 people overall, IT professionals and information management professionals, were actively involved in the evaluation process.

#### 4.2.1. Mechanisms for capturing results

The SUMI methodology (Kirakowski & Corbett, 1993) has been used to evaluate the usability of MEMOIR. In addition to this, further questionnaires have been produced within the organizations specifically to determine whether the MEMOIR functions worked as expected, and how useful the system was to researchers in practice.

It is difficult to evaluate the MEMOIR system in quantitative terms, since the objectives are not appropriate for such measurement. However, quantitative measures are appropriate for analysing uptake of services, cross-checking users' perceptions of usage with actual usage, and quantifying the robustness of the system. Such quantitative measures were achieved by analyzing the log files generated by system components.

#### 4.2.2. Results

In this section we present the results obtained from the evaluation process. The following Subsection 4.2.3 will provide an analysis of the results.

*4.2.2.1. User interface.* The evaluation process has revealed that the user interface to MEMOIR was, for some users, a barrier to the exploitation of the system's functionality. Many users have reported that the MEMOIR interface seemed out of date when compared to contemporary desktop applications, particularly users who have had exposure to the latest Web technology. Some users maintained that creating a trail was a lot of work, involving retrospective selection of items in the trail history; this is very useful feedback, since the success of MEMOIR relies on users publishing trails.

Some users have expressed privacy-related concerns. Users have questioned whether it was fair to want to know which documents their managers, or other members of their team, have accessed. Similarly, the 'Who has seen this document' functionality has been perceived in a negative way by some users, who felt that others may be able to check up on whether that person was doing their work.

*4.2.2.2. MEMOIR functionality.* The evaluation process aimed to determine whether MEMOIR functions operate as expected, and whether these functions are perceived to add value. A number of specific points relating to this issue were made in the evaluation.

Users were disconcerted that a seemingly straightforward 'create trail' request took a long time to process. It was noted that a function actively informing specific individuals in a team about a trail would be useful, rather than relying on potentially interested parties coming across the trail by generating a relevant request. A similar sentiment for notifying users of interesting documents was expressed.

From the point of view of link-related functionality, some users noted that they had to be careful with the selection of context to view information; an inappropriate choice can result in a multitude of added links in a single paragraph which do not add any value to the document. There was a favourable reaction to the citation-style of presenting links.

The keyword-related functionality was deemed useful by many evaluators. There was however, some confusion over the effect of adding further keywords to an existing query. Specifically, some feedback indicated that adding keywords appeared to widen the scope of the original query rather than narrowing the scope, as would have been expected.

There were problems encountered with network security procedures. In particular, there were very significant issues related to MEMOIR agents accessing information, as these agents did not have authorization to retrieve pages autonomously. The initial solution was to use a non-authenticated proxy server, but this became unavailable during the evaluation, and had to be overcome by the temporary measure of use of an ‘anonymous’ MEMOIR user. This is not ideal because of potential security loopholes.

Changes in results stemming from growth of the trailbase were also found confusing. As more trails are created, the statistical profile of the trailbase alters, and the results that a query returns will change. We would expect results of queries to change less frequently once a certain amount of trail data has been established. Regarding the importance of trails (and the changing thereof over time) we are planning to investigate mechanisms for degrading the relevance of trails over time. Possible approaches to address this issue include the usage of a resource based concept (see e.g., Moreau & Queinnec, 1997); or, the use of a clustering algorithm to identify and track changing interests (Crabtree & Soltysiak, 1998). By that the importance of trails can be weighed over time.

*4.2.2.3. Reaction to concepts.* The initial reaction to the MEMOIR concepts and architecture has been very positive. During the course of the evaluation most users retained the same reactions to the concepts, but were not completely satisfied with the implementation and especially the user interface.

Although the potential of the services was reported to be understood, most evaluators who were IT professionals said that they did not find the ‘fit’ between their tasks and the current set of services to be good. Other evaluators, in particular information management professionals, found the services more attractive.

*4.2.2.4. Statistics-based results.* Initial uptake of MEMOIR was high. All users logged in to MEMOIR extensively following training, but the analysis of the logs shows a gradual decline in use over time. Fig. 4 illustrates the number of pages retrieved via MEMOIR per day for the period of the month February. These results were obtained from the smaller group of evaluators (the 10 expert users).

More interesting, we also did a statistical analysis of the system services used by this group

Table 1  
Usage of MEMOIR services for expert users

Event	Occurrences
Find keywords request	28
Create trail request	46
Suggest reading for document request	57
Find person by keyword(s) request	71
Who else has seen document request	37
Create link request	4
Find documents by keyword(s) request	40

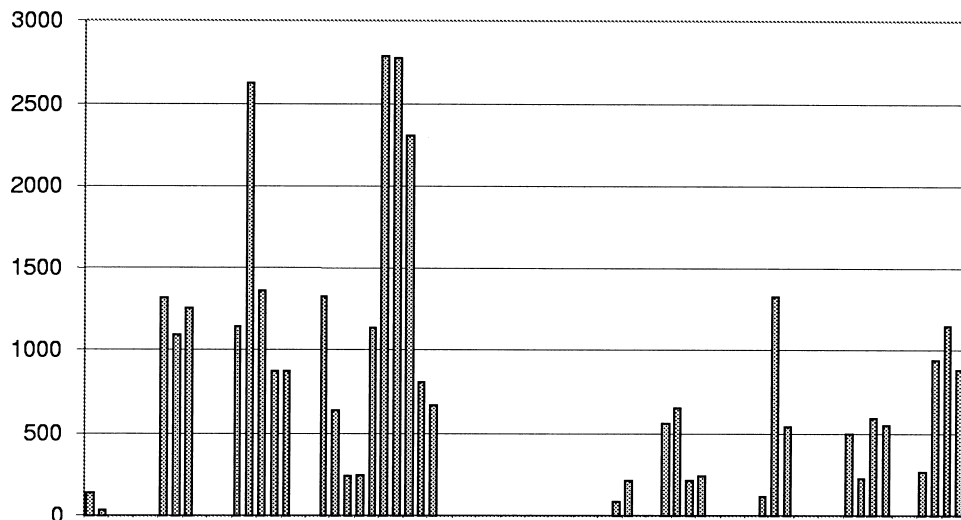


Fig. 4. Pages retrieved via MEMOIR per day over a period of one month.

of evaluators. Trail related functionality such as creation of trails or lookup of trails came first with keyword related functionality second. Table 1 illustrates these results.

For the second end user organization, an analysis of the log files shows slightly different results. These results, which have again been taken from the log files, show that keyword extraction of documents was by far the most popular functionality used. Others such as creation of trails, searching for related documents and finding users with similar interests were approximately as popular as each other. The results are shown in Table 2.

#### 4.2.3. Analysis of the results

In this section we give an analysis of the results reported in the previous section. We have grouped the analysis into user interface related issues, privacy related issues and statistics-based results.

Table 2  
Usage of MEMOIR services in day-to-day activity

Event	Occurrences
Log-in request	222
Find keywords request	491
Create trail request	61
Suggest reading for document request	57
System re-start	54
Find person by keyword(s) request	40
Who else has seen document request	40
Create link request	37
Find documents by keyword(s) request	31

*4.2.3.1. User interface related issues.* The rather unsophisticated appearance of the MEMOIR interface is due to the requirement to maintain consistency amongst a very wide range of browser platforms and operating systems. This has only been possible by avoiding the use of colour, complex formatting, and use of sophisticated GUI-based classes all of which can be rendered very differently on different platforms. Some of the compatibility problems are attributable to the rapidly evolving nature of Java and the corresponding Java Development Kit.<sup>1</sup>

With regard to the dual-window based approach to the interface, the development team has for some time recognized that alternative approaches may be preferable. Nonetheless, attempts to make the interface available as an ‘embedded applet’, appearing as a toolbar in all document browsers, were unsuccessful due to compatibility and performance issues even though the mechanisms for doing this were developed in the project (see ‘Proactive Agent’ in Section 4.1.1).

*4.2.3.2. Privacy-related issues.* The potential ‘big brother’ perception of the MEMOIR system has been recognized from an early stage; this is why a user must actively create a trail rather than the system automatically recording document access activity. It is clear that a balance must be struck between ensuring that the user recognizes that nothing is recorded without his or her consent, and minimizing the effort required to store information in the database (or even store data implicitly, see e.g., Nichols, 1997).

Nonetheless, feedback indicates that more ‘active’ publishing mechanisms would be welcomed by users. Examples are where the system offers the user the chance to publish a trail at the end of every browsing session with a single button click. If such mechanisms are available, the agents accessing the trailbase can no longer assume that all information therein is necessarily of high quality, and an automatic post-processing of automatically generated trails may be appropriate. Another facility which may address the issue is for the user to be able to publish single documents in the trailbase with a single button-click when he or she establishes the document is of interest. It is essential that the users feel they are in control of the system, and that users of the system do not perceive that automatic monitoring is in progress.

There are several important aspects related to this issue of active vs passive recommendation. Here we provide a brief analysis.

- Active recommendation offers the advantage of being able to ‘control’ the big brother problem. Only if users actually want to publish trails, they can create them.
- This process of active recommendation obviously is more work (even though the user interface may ease the publication of trails). ‘Free-riders’ (Resnick & Varian, 1997) might benefit from people investing a lot of effort in creating, managing and publishing their trails.
- Active recommendation seems to be particularly appropriate for ‘important’ or ‘new’ data, respectively. Whether some resource is new and important has to be rated by the users themselves. Something that may be new to someone might not offer new information to someone else.
- Active recommendation guarantees that groups of users share knowledge about some information between them. Again, this can be crucial in case of important information, if,

---

<sup>1</sup> The user interface of the prototype has been implemented using JDK 1.0.2.



for instance, users want to inform their colleagues about a new patent.

- Passive recommendation on the other hand, offers the big advantage of posing less effort to the users. Trails could be published automatically by extracting information from proxy logs.<sup>2</sup> This may result however, in lower quality of the trails made available to the agents.
- Related to this aspect is the issue of privacy. What type of recommendation is extracted from a published trail may not be apparent to the user at the time of publishing: the agents only later analyze the trail data and extract results. Furthermore, as has been mentioned already, users have objections about the fact that *all* their trails are published automatically.

In summary, it can be concluded that both aspects of recommendation have their advantages and disadvantages. From what we have learned in the evaluation of the MEMOIR prototype we could reason that active recommendation — if made easy by an appropriate user interface — would offer some advantages over passive recommendation.

*4.2.3.3. Statistics-based results.* The evaluation produced insufficient statistical data to enable definite trends to be discerned. Tentative indications are that people have understood the importance of publishing trails, although it is clear that many evaluators use MEMOIR for querying rather than publishing purposes (this is also referred to as ‘free riders’ (Resnick & Varian, 1997)). The keyword-based functionality seems to be particularly attractive (see Tables 1 and 2), possibly because associating keywords with documents is a concept with which users are already very familiar. Information pertaining to the selection of contexts by users was not available in the log file analysis.

## 5. Summary and conclusion

Finding people with similar interests can be as difficult as efficiently assessing documents with related information. In this paper we have reported experiences in implementing and evaluating MEMOIR, an open framework for exploiting trail based applications. We presented the idea of trails, open hypermedia link services and how software agents mine trailbases and linkbases. We have shown the open framework architecture and how users interact with the MEMOIR system. Finally we have extensively reported on results, both from re-using the open framework in research environments and from the evaluation process at two end-user organizations.

There are various conclusions to be drawn from MEMOIR.

- The reaction to the concepts underlying MEMOIR such as trails and open hypermedia linkbases has been positive. Users clearly see the potential. During the course of the

---

<sup>2</sup>This is, by the way, how the “cold start problem” was addressed in MEMOIR. We primed the trail base by taking simple data from proxy logs.

evaluation most users retained the same reactions to the concepts, but some were disappointed by the user interface of the system.

The evaluation made clear that the user interface must offer minimal hindrance to the user and has to be well integrated with the existing set of applications.

- The idea of trails will be further exploited in that different types of activities will be traced (such as printing a document) and that trails should be also generated by other applications, e.g., from an e-mail being sent with the user's favourite e-mail tool. We follow two different approaches: one uses a browser independent application, the other uses signed applets and Dynamic Data Exchange (DDE) to be better able to access user settings. Obviously both approaches are less platform independent than the current solution. A prototype following the latter approach has shown that it is useful to keep track of other activities, such as mailing a document to a colleague, and others (Downey, 1999). This prototype has not been evaluated within the MEMOIR project.
- The choice of evolving technologies such as Java must be weighed carefully where an evaluation that is dependent on presentation and robustness is planned. For instance, using Web browsers for the user interface has advantages and disadvantages. On the positive side we see the platform independence and wide availability. Also, the fact that many users are familiar with using browsers is very positive. On the negative side there are problems related to incomplete or buggy Java implementations, a constant change of tools and properties with the need of updating and the requirements of incorporating a state mechanism for asynchronous agent functionality. Furthermore, the default security model of Java is rather restrictive. Signing applets however, is different between browser platforms and even Java versions and therefore cannot be considered an option.
- The open and extensible architecture of MEMOIR in general has proven very successful. Additional or modified functionality can easily be introduced into the system by simply registering a new service with the message router. Also, using a proxy based approach for tracing user data has shown to be very successful both because of the minimal installation overhead and because of the easiness of introducing new functionality.
- The IT infrastructure into which MEMOIR has been integrated was larger, more complex and more diverse than expected. In particular, security aspects caused difficult problems and it was extremely hard to reproduce the environment externally.
- An experimental system such as MEMOIR needs to be self-contained and encapsulated, and have minimal dependencies on externally dictated factors. There should be a minimum of configuration required, and as much as possible should be automated. In addition, any configuration changes should be internal to the system, and not make demands for changes in the installation environment.

MEMOIR has been successful in many respects. It has taken some visionary ideas for a new class of IT tools supporting knowledge management processes within large globally distributed companies, and demonstrated that they can be implemented successfully. It has shown that these tools can be implemented in a way which integrates with, and is complementary to, the complex IT infrastructures of such companies. MEMOIR has also already provided a basis for new research and information systems integration projects.

## Acknowledgements

The authors would like to thank the members of the MEMOIR project, all of whom have contributed to this paper. In particular we would like to thank Andrzej Glowinski, Alastair Hotchkiss and Roger James of Glaxo Wellcome; Steve Loades and Richard Miller of Unichema; Roger Rowe from Multicosm; Eric Gerelle, Christophe Giusti and Jacques Halé of Ibex; Gerard Hutchings and Chris Scott from IT Innovation (formerly Parallel Applications Centre). Luc Moreau provided valuable feedback on an earlier draft of this paper.

We would also like to acknowledge the support of the European Union's ESPRIT programme which funded the MEMOIR project under grant No. 22153 and the EPSRC grant GR/K73060.

## References

- Andric, M., Griffiths, J. P., Reich, S., Davis, H. C., & Hall, W. (1998). Dynamically generating branching guided tours using agents and trails. In *ETRAN 98 — XLII Conference for Electronics, Telecommunications, Computers, Automation, and Nuclear Engineering, 2–5 June, Vrnjacka Banja, Yugoslavia*.
- Autonomy. (1997). Agentware i3. Technical report, Autonomy Inc., 301 University Avenue, Suite 200 Palo Alto, CA 94301, USA.
- Balabanović, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66–72.
- Barrett, R., Maglio, P. P., & Kellem, D. C. (1997). How to personalize the web. In *Conference on Human Factors in Computing Systems (CHI97), 22–27 March 1997, Atlanta, Georgia, USA* (pp. 75–82).
- Baudisch, P. (1997). The profile editor: Designing a direct manipulative tool for assembling profiles. In *Fifth DELOS Workshop, Filtering and Collaborative Filtering, Budapest, 10–12 November 1997, ERCIM report No. ERCIM-98-W001* (pp. 10–16).
- Belkin, N. J., & Croft, W. B. (1992). Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12), 29–38.
- Bentley, R., Appelt, W., Busbach, U., Hinrichs, E., Kerr, D., Sikkil, S., Trevor, J., & Woetzel, G. (1997). Basic support for cooperative work on the world wide web. *International Journal of Human-Computer Studies*, 46(6), 827–846.
- Bieber, M., Vitali, F., Ashman, H., Balasubramanian, V., & Oinas-Kukkonen, H. (1997). Forth generation hypermedia: Some missing links for the world wide web. *International Journal of Human-Computer Studies*, 47(1), 31–65.
- Bieber, M., & Wan, J. (1994). Backtracking in a multiple-window hypertext environment. In *ECHT '94 Proceedings of the ACM European conference on Hypermedia technology, 18–23 September 1994, Edinburgh, Scotland, UK* (pp. 158–166).
- Bush, V. (1945). As we may think. *The Atlantic Monthly*, 176(1), 101–108.
- Carr, L., De Roure, D., Hall, W., & Hill, G. (1995). The distributed link service: A tool for publishers, authors and readers. In *Proceedings of the Fourth International World Wide Web Conference: The Web Revolution, Boston, Massachusetts, USA*.
- Chalmers, M., Rodden, K., & Brodbeck, D. (1998). The order of things: Activity-centred information access. In *The Seventh International World Wide Web Conference, Brisbane, Australia* (pp. 359–367).
- Crabtree, I. B., & Soltysiak, S. J. (1998). Identifying and tracking changing interests. *International Journal of Digital Libraries*, 2(1), 38–53.
- Davis, H. C., Hall, W., Heath, I., Hill, G. J., & Wilkins, R. J. (1992). Towards an integrated information environment with open hypermedia systems. In *ECHT '92. Proceedings of the ACM Conference on Hypertext. 30 November–4 December 1992, Milan, Italy* (pp. 181–190).

- De Roure, D., Hall, W., Davis, H., & Dale, J. (1966). Agents for distributed multimedia information management. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK* (pp. 91–102).
- Dieberger, A. (1997). Supporting social navigation on the world-wide web. *International Journal of Human-Computer Studies*, 46(6), 805–825 (Special Issue on Innovative Applications of the Web).
- Dodge, C., Marx, B., & Pfeiffenberger, H. (1995). Web cataloguing through cache exploitation and steps toward consistency maintenance. *The Third International World-Wide Web Conference, Darmstadt, Germany, 27*, (pp. 1003–1008). Available as [http://www.igd.fhg.de/www/www95/proceedings/papers/50/AWI/Database/AW%I\\_Database.html](http://www.igd.fhg.de/www/www95/proceedings/papers/50/AWI/Database/AW%I_Database.html).
- Downey, R. T. (1999). PAMDOWN — a personal agent assisting in management of documents on the Web. Technical report, 3rd Year Project Report, Department of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK.
- Engelbart, D. C. (1963). *A conceptual framework for the augmentation of man's intellect* (vol. 1, pp. 1–29). Spartan Books London.
- Falk, A., & Jonsson, I.-M. (1996). PAWS: an agent for WWW-retrieval and filtering. In *Practical Applications of Intelligent and Multi-Agents (PAAM) '96, London* (pp. 169–179).
- FIPA. (1997). Fipa 97 specification, part 2: Agent communication language. Technical report, Foundation for Intelligent Physical Agents, Geneva, Switzerland.
- Foner, L. N. (1996). A multi-agent referral system for matchmaking. In *Practical Applications of Intelligent Agents and Multi-Agents (PAAM) '96, London* (pp. 245–261).
- Furuta, R., Shipman, F. M., Marshall, C. C., Brenner, D., & Hsieh, H.-W. (1997). Hypertext paths and the world-wide web: Experiences with walden's paths. In *Proceedings of Hypertext '97, Southampton, UK* (pp. 167–176).
- Garzotto, F., & Matera, M. (1997). A systematic method for hypermedia usability evaluation. *The new Review of Hypermedia and Multimedia*, 3, 39–65.
- Glance, N., Arregui, D., & Dardenne, M. (1997). Knowledge pump: Community-centered collaborative filtering. In *Fifth DELOS Workshop. Filtering and Collaborative Filtering. Budapest, 10–12 November, 1997. ERCIM report No. ERCIM-98-W001* (pp. 78–82).
- Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 61–70.
- Goose, S., Dale, J., Hill, G. J., De Roure, D. C., & Hall, W. (1996). An open framework for integrating widely distributed hypermedia resources. In *IEEE Conference on Multimedia and Computing Systems, Hiroshima* (pp. 364–371).
- Gottlob, G., Shrefl, M., & Röck, B. (1996). Extending object-oriented systems with roles. *ACM Transactions on Office Information Systems*, 14(3), 268–296.
- Grønbæk, K., Bouvin, N. O., & Sloth, L. (1997). Designing dexter-based hypermedia services for the world wide web. In *Proceedings of Hypertext '97, Southampton, UK* (pp. 146–156).
- Guo, H., Kreifelts, T., & Voss, A. (1997). Soap: Social filtering through social agents. In *Fifth DELOS Workshop. Filtering and Collaborative Filtering. Budapest, 10–12 November 1997. ERCIM report No. ERCIM-98-W001* (pp. 22–27).
- Hattori, F., Ohguro, T., Yokoo, M., Matusbara, S., & Yoshida, S. (1999). Socialware: Multiagent systems for supporting network communities. *Communications of the ACM*, 42(3), 55–61.
- Hill, G., Hutchings, G., James, R., Loades, S., Halé, J., & Hatzopulous, M. (1997). Exploiting serendipity amongst users to provide support for hypertext navigation. In *Proceedings of Hypertext '97, Southampton, UK* (pp. 212–213).
- Hill, G., Wilkins, R., & Hall, W. (1993). Open and reconfigurable hypermedia systems: A filter based model. *Hypermedia*, 5(2), 103–118.
- Huberman, B. A., & Kaminsky, M. (1996). Beehive: A system for cooperative filtering and sharing of information. Technical report, Dynamics of Computation Group, Xerox Palo Alto Research Center.
- Huhns, M. N., & Singh, M. P. (1997). Agents on the web. *IEEE Internet Computing*, 1(5), 78–79.
- IBEX. (1995). *ITASCA<sup>TM</sup> Distributed Object Database Management System. Release 2.3.5, Technical Summary*. IBEX Corporation SA, F-74160 Archamps, France.

- Joachims, T., Freitag, D., & Mitchell, T. (1997). Webwatcher: A tour guide for the world wide web. In *International Joint Conference on Artificial Intelligence (IJCAI-97)*, Nagoya, Aichi, Japan.
- Jones, S., & Cockburn, A. (1996). A study of navigational support provided by world wide web browsing applications. In *Proceedings of the '96 ACM Conference on Hypertext, 16–20 March 1996, Washington, DC* (pp. 161–169).
- Jühne, J., Jensen, A. T., & Grønbæk, K. (1998). Ariadne: a java-based guided tour system for the world wide web. In *The Seventh International World Wide Web Conference, Brisbane, Australia* (pp. 131–139).
- Kahle, B., & Gilliat, B. (1997). Alexa — navigate the Web smarter, faster, easier. Technical report, Alexa Internet, Presidio of San Francisco, CA. Available as <http://www.alexa.com/>.
- Keller, R. M., Wolfe, S. R., Chen, J. R., Rabinowitz, J. L., & Mathe, N. (1997). A book-marking service for organizing and sharing URLs. In *The Sixth International World Wide Web Conference, Santa Clara, CA*.
- Kirakowski, J., & Corbett, M. (1993). SUMI: The software usability measurement inventory. *British Journal of Educational Technology*, 24(3), 210–212.
- Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3), 77–87.
- Labrou, Y., & Finin, T. (1997). A proposal for a new KQML specification. Technical Report TR CS-97-03, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, Baltimore, MD 21250.
- Lieberman, H. (1995). Letizia: An agent that assists web browsing. In *International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada (pp. 924–929).
- Loades, S., James, R., Hutchings, G., & Stairmand, M. (1997). D2.3 report on refined user requirements. Technical Report UIC/MEMOIR/D2.3, ESPRIT Programme, MEMOIR Project (22153).
- Malcolm, K. C., Poltrock, S. E., & Schuler, D. (1991). Industrial strength hypermedia: Requirements for a large engineering enterprise. In *Proceedings of the '91 ACM Conference on Hypertext, 15–18 December 1991, San Antonio, TX* (pp. 13–24).
- Maltz, D., & Ehrlich, K. (1995). Pointing the way: active collaborative filtering. In *Conference on Human Factors in Computing Systems (CHI95)*, 7–11 May 1995 Denver, Colorado, USA (pp. 202–209).
- Moreau, L., & Queinnec, C. (1997). Design and semantics of Quantum: a language to control resource consumption in distributed computing. In *Usenix Conference on Domain-Specific Languages (DSL '97)*, Santa-Barbara, California (pp. 183–197).
- Nichols, D. M. (1997). Implicit rating and filtering. In *Fifth DELOS Workshop. Filtering and Collaborative Filtering. Budapest, 10–12 November 1997. ERCIM report No. ERCIM-98-W001* (pp. 28–33).
- Nicol, D., Smeaton, C., & Slater, A. F. (1995). Footsteps: Trail-blazing the web. *The Third International World-Wide Web Conference. Darmstadt, Germany, 27* (pp. 879–885). Published in Computer Networks and ISDN Systems.
- Nwana, H. S., & Ndumu, D. T. (1996). An introduction to agent technology. *BT Technology Journal*, 14(4), 55–67.
- Petrie, C. J. (1997). What's an agent ... and what's so intelligent about it? *IEEE Internet Computing*, 1(4).
- Pikrakis, A., Bitsikas, T., Sfakianakis, S., Hatzopoulos, M., De Roure, D. C., Hall, W., Reich, S., Hill, G. J., & Stairmand, M. (1998). Memoir — software agents for finding similar users by trails. In *PAAM98. The Third International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, 23–25 March, London, UK* (pp. 453–466).
- Plaisant, C., Shneiderman, B., & Mushlin, R. (1998). An information architecture to support the visualization of personal histories. *Information Processing and Management*, 34(5), 581–597.
- Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3), 56–58.
- Röscheisen, M., Mogensen, C., & Winograd, T. (1995). Beyond browsing: Shared comments, soaps, trails, and on-line communities. *The Third International World-Wide Web Conference, Darmstadt, Germany, 27* (pp. 739–749). Published in Computer Networks and ISDN Systems.
- Rosenberg, J. (1996). The structure of hypertext activity. In *Proceedings of the '96 ACM Conference on Hypertext, 16–20 March 1996, Washington, DC* (pp. 22–29).
- Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for automating “word of mouth”. In *Conference on Human Factors in Computing Systems (CHI95)*, 7–11 May 1995, Denver, Colorado, USA (pp. 210–217).

- Shipman, F. M., Furuta, R., Brenner, D., Chung, C.-C., & Hsieh, H.-W. (1998). Using paths in the classroom: Experiences and adaptations. In *Proceedings of the '98 ACM Conference on Hypertext, 20–24 June 1998, Pittsburgh, PA* (pp. 267–276).
- Shneiderman, B., & Maes, P. (1997). Direct manipulations vs interface agents. *Interactions*, 4(6).
- Starr, B., Ackerman, M. S., & Pazzani, M. (1996). Do-I-Care: A collaborative web agent. In *Conference on Human Factors in Computing Systems (CHI96), 13–18 April 1996, Vancouver, Canada* (pp. 273–274).
- Tauscher, L., & Greenberg, S. (1997). Revisitation patterns in world wide web navigation. In *Conference on Human Factors in Computing Systems (CHI97), 22–27 March 1997, Atlanta, Georgia, USA* (pp. 399–406).
- Terveen, L., Hill, W., Amento, B., McDonald, D., & Creter, J. (1997). Phoaks: A system for sharing recommendations. *Communications of the ACM*, 40(3), 59–62.
- Tudhope, D., & Taylor, C. (1997). Navigation via similarity: Automatic linking based on semantic closeness. *Information Processing and Management*, 33(2), 233–242.
- Twidale, M. B., Nichols, D. M., & Paice, C. D. (1997). Browsing is a collaborative process. *Information Processing and Management*, 33(6), 761–783.
- Tympani. (1997). Tympani Atlas Server. Technical report, Tympani Development Inc., 1307 South Mary Ave., Suite 102 Sunnyvale, CA 94087. Available as <http://www.tympani.com/products/ATLAS/AtlasBrochure.pdf>.
- Whitehead, J. E. (1997). World wide web distributed authoring and versioning (WEBDAV): An introduction. *ACM StandardView*, 5(1), 3–8.
- Wiil, U. K. (1997). Open hypermedia: Systems, interoperability and standards. *Journal of Digital Information (JoDI)*, 1(2) Special Issue on Open Hypermedia.
- Wiil, U. K. (1999). *Proceedings of the 5th Workshop on Open Hypermedia Systems, Hypertext '99, Darmstadt, Germany, 21–25 February*.
- Wooldridge, M., & Jennings, N. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2), 115–152.
- Zellweger, P. T. (1989). Scripted documents: A hypermedia path mechanism. In *Proceedings of the '89 ACM Conference on Hypertext, 5–9 November 1989, Pittsburgh, PA* (pp. 1–14).
- Zuno. (1997). Vrisko — personal knowledge manager. Technical report, Zuno Limited, 300 Third Avenue, Waltham, MA 02154. Available as <http://www.dlib.com/Products/vrisko.html>.